METHOD AND APPARATUS FOR MULTI-PROTOCOL REDUNDANT ROUTER PROTOCOL SUPPORT

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims the priority of U.S. Provisional Application No. 60/206,617 entitled "System and Method for Enhanced Line Cards" filed May 24, 2000, U.S. Provisional Application No. 60/206,996 entitled "Flow Resolution Logic System and Method" filed May 24, 2000, U.S. Provisional Application No. 60/220,335 entitled "Programmable Packet Processor" filed July 24, 2000 and U.S. Provisional Application No. 60/232,479 entitled "Hot Standby Routing" filed September 13, 2000, the contents of all of which are fully incorporated by reference herein. The present application contains subject matter related to the subject matter disclosed in U.S. Patent Application No. 09/751,194 entitled "Programmable Packet Processor with Flow Resolution Logic" filed December 28, 2000, the contents of which are fully incorporated by reference herein.

FIELD OF THE INVENTION

The present invention is related to redundant routing, and particularly to a method and apparatus for providing multi-protocol redundant router protocol support.

BACKGROUND OF THE INVENTION

Redundant routing protocols have been developed to provide hosts configured with static routes a measure of protection against router failure. In redundant routing, a host is configured to send to a virtual router MAC address that is supported by two or more physical routers sharing a

LAN with the host. Particularly, at any given time in an operational cycle, one of the physical routers, a virtual master, is responsible for forwarding packets received from the host and having the virtual router MAC address, and the other backup routers standby to assume forwarding responsibilities in the event the virtual master fails. The transition by which respective ones of the backup routers become the virtual master is transparent to the host.

In addition to their failure recovery characteristics, redundant routing protocols can be used advantageously in LANs having two or more hosts to achieve load sharing. In a load sharing arrangement, at least two hosts are assigned different ones of virtual router MAC addresses such that different ones of the physical routers become the initial virtual master for the different ones of the hosts.

While redundant router protocols have clear advantages, adding redundant router protocol hardware support to routers is typically expensive. Additional caching facilities are typically required on the participating physical routers to store the 48-bit MAC addresses for the active virtual routers. This implementation cost has been exacerbated by the existence of two competing (and non-interoperable) redundant routing protocols: Hot Standby Router Protocol (HSRP), specified in Internet Engineering Task Force (IETF) Request for Comment (RFC) 2281 and Virtual Router Redundancy Protocol (VRRP) specified in IETF RFC 2338.

Therefore, it is desirable to provide efficient redundant router protocol support in general, and multi-protocol redundant router protocol support, in particular.

SUMMARY OF THE INVENTION

In one embodiment of the present invention, a local area network (LAN) is provided. The LAN includes a plurality of hosts, a plurality of physical routers and a LAN medium interconnecting the hosts and the physical routers. A first one of the hosts applies a packet of a first redundant router protocol type to the LAN medium and a second one of the hosts applies a packet of a second redundant router protocol type to the LAN medium. The physical routers determine responsibility for forwarding a packet received on the LAN medium in function of a redundant router protocol type of the packet.

In another embodiment of the present invention, a method of routing a plurality of packets using a plurality of redundant routing protocols is provided. A router receives a packet having a packet address. A prefix of the packet address is compared with a first predefined value to determine whether the packet is of a first redundant routing protocol type. The prefix of the packet address is compared with a second predefined value to determine whether the packet is of a second redundant routing protocol type.

In yet another embodiment of the present invention, a router for receiving and forwarding one or more packets is provided. The router includes a first comparator for comparing a packet address prefix and a first predefined value to determine whether the packet is of a first redundant router protocol type. The router also includes a second comparator for comparing the packet address prefix and a second predefined value to determine whether the packet is of a second redundant router protocol type.

BREIF DESCRIPTION OF THE DRAWINGS

These and other aspects of the invention may be understood by reference to the following detailed description, taken in conjunction with the accompanying drawings, which are briefly described below.

FIG. 1 is a system diagram of an apparatus for supporting both the HSRP and VRRP protocols according to an embodiment of the present invention;

FIG. 2 illustrates a network environment including a packet switching node, such as a router, according to an embodiment of the present invention;

FIG. 3 is a block diagram of a switching interface according to an embodiment of the present invention;

FIG. 4 is a packet switching controller according to an embodiment of the present invention;

FIG. 5 is a schematic diagram illustrating a process of determining whether the incoming data unit is of the type HSRP or VRRP according to an embodiment of the present invention; and

FIG. 6 is a flow diagram illustrating a process of determining whether the incoming data unit is of the type HSRP or VRRP according to an embodiment of the present invention.

DETAILED DESCRIPTION

FIG. 1 is a system diagram of an apparatus for supporting both HSRP and VRRP protocols according to an embodiment of the present invention. In FIG. 1, a local area network (LAN) includes a plurality of hosts 100, 102, 104, 106 and a plurality of routers 110, 116, which are physical (as opposed to virtual) routers. The routers 110 and 116 are coupled to a computer network 120. The routers

110, 116 may be viewed as being coupled to the LAN to provide gateway to the computer network 120. In other embodiments, the routers 110, 116 may be coupled to one or more LANs other than the LAN of FIG. 1.

The computer network 120, for example, may include the Internet or other global or local computer networks. The routers 110 and 116 may also be coupled to one or more other LANs (not shown). The LANs in this and other embodiments may have one or more different configurations including, but not limited to, Ethernet (IEEE 802.3), token ring (IEEE 802.5) and FDDI (ANSI X3T9.5).

The hosts 100 and 104 preferably are associated with a group of redundant routers HSRP Group 1 and HSRP Group 2, respectively. The hosts 102 and 106 are associated with a group of redundant routers VRRP Group1 and VRRP Group 2, respectively. It should be noted that the hosts and routers of FIG. 1 are shown for illustrative purposes only. In practice, the LAN may include one or more additional hosts and routers belonging to HSRP Groups 1 and/or 2, VRRP Groups 1 and/or 2, and/or other HSRP and/or VRRP groups. The redundant routers in each group share a common virtual router address, which is assigned to one or more hosts associated with the group of redundant routers.

The virtual router addresses may include a Media Access Control (MAC) address, a network address (e.g., IP address) or both. When the host assigned to a virtual router address transmits one or more data units (e.g., Ethernet frames, IP packets or ATM cells) to be routed, the data units are directed to one of the redundant routers in the group that is acting as the virtual master for that particular group.

For example, data units from the host 100 preferably are routed by an HSRP Group 1 virtual router 108, data units from the host 102 preferably are routed by a VRRP Group 1 virtual router 118, data units from the host 104 preferably are routed by an HSRP Group 2 virtual router 114, and data units from the host 106 preferably are routed by a VRRP Group 2 virtual router 112.

Since the HSRP and VRRP virtual routers 108, 112, 114 and 118 are not physical routers, their virtual router addresses preferably are mapped to the routers 110 and 116. For example, in an embodiment according to the present invention, the router 110 preferably is configured as an HSRP Group 1 virtual master and a VRRP Group 2 virtual master. For another example, the router 116 is configured as a VRRP Group 1 virtual master and an HSRP Group 2 virtual master. Virtual masters may also be referred to as active routers, virtual router masters or Masters.

In FIG. 1, the routers 110 and 116 are illustrated to be supporting four groups of virtual routers (i.e., HSRP Group 1 virtual router, VRRP Group 2 virtual router, HSRP Group 2 virtual router, and VRRP Group 1 virtual router). Therefore, for example, when the router 110 operates as the HSRP Group 1 virtual master and the VRRP Group 2 virtual master, the router 116 may operate as an HSRP Group 1 standby router and a VRRP Group 2 standby router. For another example, when the router 116 operates as the HSRP Group 2 virtual master and the VRRP Group 1 virtual master, the router 110 may operate as an HSRP Group 2 standby router and a VRRP Group 1 standby router. Standby routers may also be referred to as backup routers. In other embodiments, each physical router may be mapped to one HSRP

group of redundant routers and one VRRP group of redundant routers.

In practice, for example, each of the routers 110 and 116 may support up to four HSRP virtual router groups and up to four VRRP virtual router groups simultaneously on up to 512 different LANs. In other embodiments, each of the routers 110 and 116 may support more than four HSRP virtual router groups and more than four VRRP virtual router groups simultaneously on 512 or more different LANs.

In FIG. 2, a network environment including a packet switching node 120 is illustrated. The packet switching node 120, for example, may be used as one or both of the routers 110 and 116. The packet switching node 120 includes a number of switching interfaces 124, 126, 128 preferably interconnected to respective groups of LANs 130, 132, 134 and preferably interconnected to each other over data paths 138, 140, 142 via a switching backplane 122 and over control paths 144, 146.

The switching interfaces 124, 126, 128 preferably forward packets to and from their respective groups of LANs 130, 132, 134 in accordance with one or more operative communication protocols, such as, for example, media access control (MAC) bridging and Internet Protocol (IP) routing. The switching interfaces 124, 126 and 128 preferably communicate with other packet switching nodes over a computer network 136, which may include the Internet and/or other global or local computer networks.

FIG. 3 is a block diagram of a switching interface 150, which may be similar to one or more of the switching interfaces 124, 126 and 128. The switching interface 150 includes an access controller 154 coupled between the LANs and a packet switching controller 152. The access

controller 154 preferably receives inbound packets off LANs, performs flow-independent physical and MAC layer operations on the inbound packets and transmits the inbound packets to the packet switching controller 152 for flow-dependent processing. The access controller 154 preferably also receives outbound packets from the packet switching controller 152, preferably performs physical and MAC layer operations on the outbound packets and transmits the outbound packets on the LANs or to a computer network, such as, for example, the computer network 136 of FIG. 2.

The packet switching controller 152 preferably receives inbound packets, classifies the packets, generates application data for the inbound packets, modifies the inbound packets in accordance with the application data, and transmits the modified inbound packets on a switching backplane, such as, for example, the switching backplane 122 of FIG. 2. The packet switching controller 152 preferably also receives outbound packets from other packet switching controllers over the backplane, and transmits the outbound packets to the access controller 154 for forwarding on the LANs or to the computer network, such as, for example, the compute network 136 of FIG. 2. In other embodiments, the packet switching controller 152 may also subject one or more outbound packets to egress processing prior to forwarding them to the access controller 154. The packet switching controller 152 may be implemented in non-programmable logic, programmable logic or any combination of programmable and non-programmable logic.

FIG. 4 is a block diagram of a programmable packet switching controller 200 according to an embodiment of the present invention. The programmable packet switching controller 200, for example, may be similar to the packet

switching controller 152 of FIG. 3. The programmable packet switching controller 200 preferably has flow resolution logic for classifying and routing incoming flows of packets. Packet switching controllers in other embodiments may include more or less number of components. For example, a packet switching controller in another embodiment may include a pattern match module for comparing packet portions against a predetermined pattern to look for a match. The packet switching controller in yet another embodiment may include an edit module for editing inbound packets to generate outbound packets. Further, packet switching controllers in still other embodiments may include other components, such as, for example, a policing engine, in addition to or instead of the components included in the programmable packet switching controller 200.

Due to its programmable nature, the programmable packet switching controller 200 preferably provides flexibility in handling many different protocols and/or field upgradeability. The programmable packet switching controller 200 may also be referred to as a packet switching controller, a switching controller, a programmable packet processor, a network processor, a communications processor or as another designation commonly used by those skilled in the art.

The programmable packet switching controller 200 includes a packet buffer 202, a packet classification engine 204, and an application engine 206. The programmable packet switching controller 200 preferably receives inbound packets 208. The packets (or data units) may include, but are not limited to, Ethernet frames, ATM cells, TCP/IP and/or UDP/IP packets, and may also include

other Layer 2 (Data Link/MAC Layer), Layer 3 (Network Layer) or Layer 4 (Transport Layer) data units. For example, the packet buffer 202 may receive inbound packets from one or more Media Access Control (MAC) Layer interfaces over the Ethernet.

The received packets preferably are stored in the packet buffer 202. The packet buffer 202 may include a packet FIFO for receiving and temporarily storing the packets. The packet buffer 202 preferably provides the stored packets or portions thereof to the packet classification engine 204 and the application engine 206 for processing.

The packet buffer 202 may also include an edit module for editing the packets prior to forwarding them out of the switching controller as outbound packets 218. The edit module may include an edit program construction engine for creating edit programs real-time and/or an edit engine for modifying the packets. The application engine 206 preferably provides application data 216, which may include a disposition decision for the packet, to the packet buffer 202, and the edit program construction engine preferably uses the application data to create the edit programs. The outbound packets 218 may be transmitted over a switching fabric interface to communication networks, such as, for example, the Ethernet.

The packet buffer 202 may also include either or both a header data extractor and a header data cache. The header data extractor preferably is used to extract one or more fields from the packets, and to store the extracted fields in the header data cache as extracted header data. The extracted header data may include, but are not limited to, some or all of the packet header. In an Ethernet

system, for example, the header data cache may also store first N bytes of each frame.

The extracted header data preferably is provided in an output signal 210 to the packet classification engine 204 for processing. The application engine may also request and receive the extracted header data over an interface 214. The extracted header data may include, but are not limited to, one or more of Layer 2 MAC addresses, 802.1P/Q tag status, Layer 2 encapsulation type, Layer 3 protocol type, Layer 3 addresses, ToS (type of service) values and Layer 4 port numbers. In other embodiments, the output signal 210 may include the whole inbound packet, instead of or in addition to the extracted header data. In still other embodiments, the packet classification engine 204 may be used to edit the extracted header data to be placed in a format suitable for use by the application engine, and/or to load data into the header data cache.

The packet classification engine 204 preferably includes a programmable microcode-driven embedded processing engine. The packet classification engine 204 preferably is coupled to an instruction RAM (IRAM) (not shown). The packet classification engine preferably reads and executes instructions stored in the IRAM. In one embodiment, many of the instructions executed by the packet classification engine are conditional jumps. In this embodiment, the classification logic includes a decision tree with leaves at the end points that preferably indicate different types of packet classifications. Further, branches of the decision tree preferably are selected based on comparisons between the conditions of the instructions and the header fields stored in the header data cache. In

other embodiments, the classification logic may not be based on a decision tree.

In one embodiment of the present invention, the application engine 206 preferably has a pipelined architecture wherein multiple programmable sub-engines are pipelined in series. Each programmable sub-engine preferably performs an action on the packet, and preferably forwards the packet to the next programmable sub-engine in a "bucket brigade" fashion. The packet classification engine preferably starts the pipelined packet processing by starting the first programmable sub-engine in the application engine using a start signal 212. The start signal 212 may include identification of one or more programs to be executed in the application engine 206. The start signal 212 may also include packet classification information. The programmable sub-engines in the application engine preferably have direct access to the header data and the extracted fields stored in the header data cache over the interface 214.

The application engine may include other processing stages not performed by the programmable sub-engines, however, the decision-making stages preferably are performed by the programmable sub-engines to increase flexibility. In other embodiments, the application engine may include other processing architectures.

FIG. 5 is a schematic diagram illustrating a process of determining whether the incoming data unit is of the type HSRP or VRRP, according to an embodiment of the present invention. The schematic diagram of FIG. 5 includes a prefix match block 250 and a database table 252. The prefix match block 250 may be included in a packet classification engine, such as, for example, the packet

classification engine 204 of FIG. 4. In other embodiments, the prefix match block 250 may be included in an application engine, such as, for example, the application engine 206 of FIG. 4.

The database table 254 preferably includes a bit table, which is indexed by protocol selection (HSRP or VRRP), VLAN number (or VLAN ID/address) and group number (or group ID) to yield a single bit result. In other embodiments, the database table 254 may be in other table format and other parameters may be used to index the table. The result yielded by the database table 254 may include multiple bits in other embodiments. The database table 254 may be included in the application engine or it may be implemented in memory external to the application engine.

FIG. 5 may be described in reference to FIG. 6. FIG. 6 is a flow diagram illustrating a process of determining whether the incoming data unit is of the type HSRP or VRRP according to an embodiment of the present invention. In step 302, the process receives a data unit. The data unit includes a destination address, which may include a destination Media Access Control (DMAC) address an/or a Virtual Local Area Network (VLAN) ID. An exemplary DMAC address and VLAN ID illustrated in FIG. 5 contains 48 bits and 12 bits, respectively.

In step 303, the prefix match block 250 preferably is used to determine whether the received data unit is of the type HSRP or VRRP. Each of the HSRP and VRRP router MAC addresses includes an IEEE 802 MAC address, and has pre-defined 40-bit prefix with an 8-bit group suffix. For example, the virtual MAC address for an HSRP group may be 00-00-0C-07-AC-XX, where each of 00, 0C, 07 and AC is an 8-bit hexadecimal number and XX is an 8-bit group ID for the

HSRP group. For another example, the virtual MAC address for an VRRP group may be 00-00-5E-00-01-XX, wherein each of 00, 5E and 01 is an 8-bit hexadecimal number and XX is an 8-bit group ID for the VRRP group. If the first 40 bit prefix of the received DMAC matches 40-bit prefix for neither HSRP nor VRRP, the process indicates no prefix match in a decision 304. If there is no prefix match, the data unit may not have been directed to one of the virtual routers.

In step 305, the prefix match block 250 preferably also checks the VLAN ID and the VRRP/HSRP group ID of the received data unit to determine whether they are within a predetermined range of values. For example, in the embodiment where a router may support up to 512 different LANs simultaneously, the value of the VLAN ID should be between 0 and 511, inclusive. For another example, in the embodiment where a router may support up to four HSRP groups and/or four VRRP groups simultaneously, the value of the VRRP/HSRP group ID should be between 0 and 3, inclusive. If either the VLAN ID or the VRRP/HSRP group ID is not within their respective range of values, the process indicates out of range in a decision 306. If either the VLAN ID or the VRRP/HSRP group ID is out of range, the data unit may be routed using software, but typically at a slower rate.

In other embodiments, additional number of different LANs and/or additional number of VRRP/HSRP groups may be supported. For example, up to 4096 different LANs may be supported using all 12 bits of a 12-bit VLAN ID and up to 256 VRRP/HSRP groups may be supported using all 8 bits of an 8-bit group ID.

In step 308, the prefix match block 250 preferably formats a key for matching in the database table 252. In the exemplary embodiment illustrated in FIG. 5, the key contains 12 bits. In other embodiments, the key may include more or less number of bits than 12. The key preferably includes a protocol ID 254, a VLAN address 256 and a group ID 258. The protocol ID 254 preferably is a single bit identification of either HSRP or VRRP. The VLAN address 256 preferably includes nine least significant bits (LSBs) of the 12-bit VLAN ID for the received data unit. The group ID preferably includes two bits to signify either the VRRP or HSRP group ID.

Using the 12-bit key, the router including the prefix match block of the described embodiment is capable of processing HSRP and VRRP packets concurrently with up to 4 groups of each type on up to 512 VLANs. In other embodiments, the number of bits in the protocol ID 254, the VLAN address 256 and/or the group ID 258 may be different, and may result in different HSRP and/or VRRP packet processing capabilities. In still other embodiments, the number of bits in the protocol ID, the VLAN address and/or the group ID may be programmable to support different number of redundant router protocols (e.g., protocols other than HSRP and VRRP may be defined in the future) and/or virtual router addresses.

In step 310, the key preferably is compared against one or more entries in the database table 252. The data base table may include VRRP/HSRP database, which may also be referred to as VRRP/HSRP virtual master database. If the key does not match any of the entries in the database table 252, the router containing the database table is not operating (314) as the virtual master for the host that

transmitted the data unit. If the key matches an entry in the database table, a match bit is generated to indicate that the router is operating as the virtual master. The data unit preferably is routed (or switched) using the virtual router address when the key matches, as indicated in step 316.

It will be appreciated by those of ordinary skill in the art that the invention can be embodied in other specific forms without departing from the spirit or essential character hereof. The present description is therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims, and all changes that come within the meaning and range of equivalents thereof are intended to be embraced therein.

For example, the described embodiments of the present invention have been described in reference to use of multi-protocol redundant router protocol support in programmable packet switching controllers. However, the multi-protocol redundant router protocol support of the present invention may also be applied to non-programmable, e.g., hard-wired, packet switching controllers.